Quesito D1: (4p)

L'algoritmo Shortest Execution Time First (SETF), o Shortest Job First, esegue i task attribuendo la priorità in funzione del tempo di esecuzione dei task (priorità massima al task con il minor tempo di esecuzione). Questo criterio di priorità tende a consentire l'esecuzione di un maggior numero di task in un tempo assegnato, e a ridurre i tempi medi di attesa. Dimostrare se SETF è oppure non è ottimo per la schedulazione in tempo reale di task periodici la cui deadline relativa coincide con il periodo.

Quesito D2: (8 p)

- a) Dato l'insieme di task periodici indipendenti *schedulati in modo RM*: TS=[τ 1=(5,1.6), τ 2=(6,1.2), τ 3=(10,2.8), τ 4=(15,1.6)] (tutti i parametri sono espressi in ms), determinare, argomentando appropriatamente la risposta, se **l'insieme di task è schedulabile** (si, no, forse), specificando in tutti i casi quali **task sono individualmente garantiti** (si, no), in base ai seguenti criteri:
- il bound di Liu e Layland
- il bound di Kuo e Mok
- il bound iperbolico.
- b) Verificare inoltre se il task a priorità minima è garantito mediante calcolo del tempo di risposta utilizzando una delle formulazioni algebriche dell'algoritmo di Audsley.
- c) Si supponga infine che i task τ i presentino le seguenti sezioni di codice non revocabili NRi: NR1=0.2, NR2=1, NR3=1.5, NR4=0.2 (durate espresse in ms). Determinare quali task risultano essere garantiti in base ai bound di Liu e Layland e Kuo e Mok.

Quesito D3: (8 p)
Dimostrare che l'algoritmo RM è ottimo tra gli algoritmi di schedulazione real-time a priorità statica per task la cui deadline relativa coincide con il periodo.

(Argomentate la dimostrazione anche con parole a commento, dimostrando di averla compresa.)

Quesito D4: (4 p)
Presentare l'algoritmo EDD (Earliest Due Date). Discuterne le ipotesi e il campo di applicazione e riportare il test di garanzia che deve essere applicato per valutare la schedulabilità di un insieme di N task indipendenti.

Quesito D5: (4 p)

Svolgere le analisi preliminari, le formule e i calcoli nello spazio bianco in fondo, poi riportare il risultato finale nelle griglie predisposte..

Si consideri un insieme di 5 task periodici interagenti τ 1, τ 2, τ 3, τ 4, τ 5 in esecuzione su un sistema che supporta la full preemption e l'opzione tra i protocolli *Priority Inheritance (PIP)* oppure *Non-Preemptive Critical Sections (NPCS)*. Si supponga che tra i task siano presenti le sezioni critiche C1, C2, C3, C4, protette da semafori mutex distinti e prive di annidamenti. Le priorità dei task sono di tipo statico ed attribuite in ordine decrescente, con τ 1 a massima priorità.

Gli accessi alle sezioni critiche da parte dei task sono i seguenti:

- τ1 accede a C1 con durata 10 ms, a C3 con durata 8 ms, a C2 con durata 12 ms.
- τ2 accede a C2 con durata 10 ms, a C3 con durata 10 ms, ancora a C2 con durata 21 ms.
- τ3 accede a C1 con durata 16 ms, a C4 con durata 14 ms.
- τ4 accede a C1 con durata 10 ms, a C2 con durata 7 ms, a C4 con durata 48 ms.
- τ5 accede a C4 con durata 97 ms, a C1 con durata 1 ms, a C3 con durata 6 ms.

Calcolare il <u>tempo totale di blocco di caso peggiore Bi</u> (ovvero i tempi di inversione di priorità) ed il <u>numero massimo di situazioni di inversione di priorità Ni</u> che ciascun task può subire con il protocollo **PIP** e quindi da considerare per l'analisi di schedulabilità. Riportare le risposte nella tabella sottostante.

B1	N1	B2	N2	В3	N3	B4	N4	B5	N5

Con riferimento al medesimo insieme di task, si supponga ora che il protocollo adottato per la prevenzione della inversione illimitata di priorità sia *Non-Preemptive Critical Sections* (NPCS). Si riportino nella tabella sottostante i corrispondenti valori di tempo totale di blocco di caso peggiore Bi e numero massimo di situazioni di inversioni di priorità Ni per ogni task.

B1	N1	B2	N2	В3	N3	B4	N4	B5	N5

Quesito D6: (4 p)

In un sistema di elaborazione in tempo reale è essenziale garantire l'esecuzione di un unico task sporadico τ_r con minimo tempo di interarrivo superiore a 331 ms, tempo di esecuzione massimo di 5 ms, deadline relativa pari a 207 ms. Il sistema deve inoltre garantire l'esecuzione del seguente **insieme di task periodici mediante scheduling priority-driven statico**: [τ 1=(5,1), τ 2=(8,2), τ 3=(10,2), τ 4=(16,2)] (i parametri sono espressi in ms). Il task τ_r deve essere gestito mediante **polling server**. Si proponga una soluzione che soddisfi le specifiche, fornendo i relativi parametri di schedulazione e dimensionando opportunamente il polling server. Tra le soluzioni formalmente possibili si identifichi quella che determina il minimo overhead per il sistema complessivo.